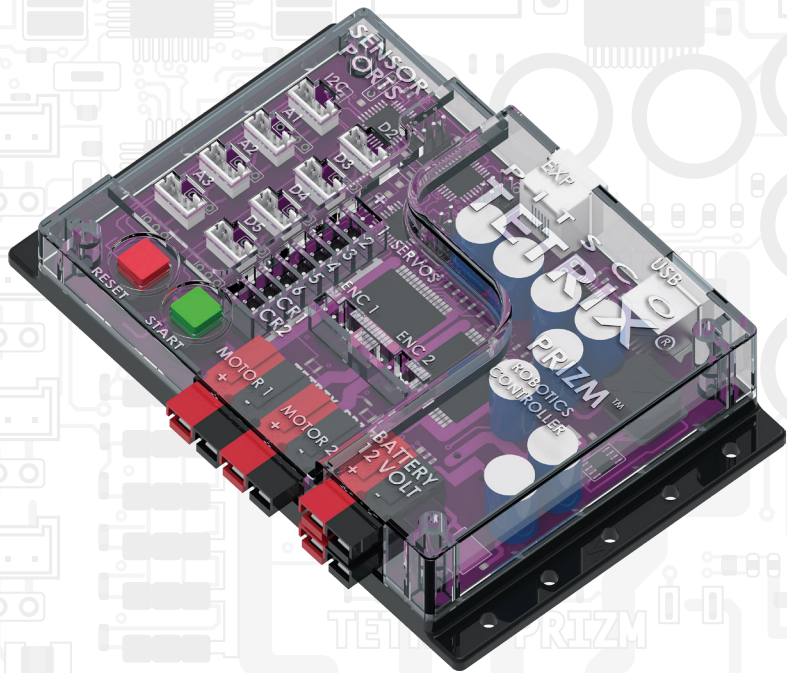


PITSCO

TETRIX
MAX



TETRIX® PRIZM™ Robotics Controller Quick-Start Guide and Specifications

PITSCO
EDUCATION

43167

Content advising by Paul Uttley.

SolidWorks® Composer™ renderings by Tim Lankford and Brian Eckelberry.

Desktop publishing by Todd McGeorge.

©2016 Pitsco, Inc., 915 E. Jefferson, Pittsburg, KS 66762

All rights reserved. This product and related documentation are protected by copyright and are distributed under licenses restricting their use, copying, and distribution. No part of this product or related documentation may be reproduced in any form by any means without prior written authorization of Pitsco, Inc.

All other product names mentioned herein might be the trademarks of their respective owners.

A downloadable PDF of the most recent version of this guide can be found at www.TETRIXrobotics.com/PRIZMdownloads.

This device complies with Part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) this device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

California Prop 65 - **WARNING:** This product contains one or more chemicals known to the State of California to cause cancer, birth defects, or other reproductive harm.

V1.2
09/16

TETRIX® PRIZM™ Robotics Controller

Quick-Start Guide and Specifications

General Description

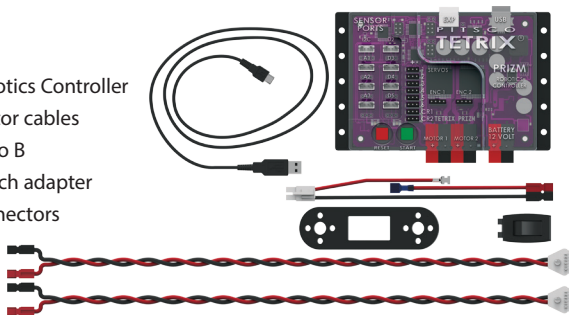
TETRIX® PRIZM™ is a programmable target designed to enable autonomous control of robotics and mechatronic systems constructed using TETRIX MAX building elements. The design architecture of PRIZM is Arduino compatible, using the same ATmega328P program memory chip and preprogrammed Optiboot bootloader as a genuine Arduino UNO. PRIZM offers a fully integrated brain for your robot that is powerful yet easy to use and, when combined with the *Arduino Software (IDE)*, can provide an ideal learning tool for teaching engineering design and coding.

The TETRIX PRIZM Robotics Controller features the following:

- 2 high-current DC motor control ports, 6 standard control servo ports, and 2 continuous rotation (CR) servo ports
- 4 digital sensor ports (D2 can be used as a serial port; D2-D5 can be configured as digital input or output) and 3 analog sensor ports (A1-A3 can be configured as analog input or digital input or output)
- 1 I2C port, 1 USB programming port, 1 motor controller expansion port, and 2 quadrature encoder input ports
- 2 battery connection ports (additional port used to daisy-chain power to other devices)

What's Included:

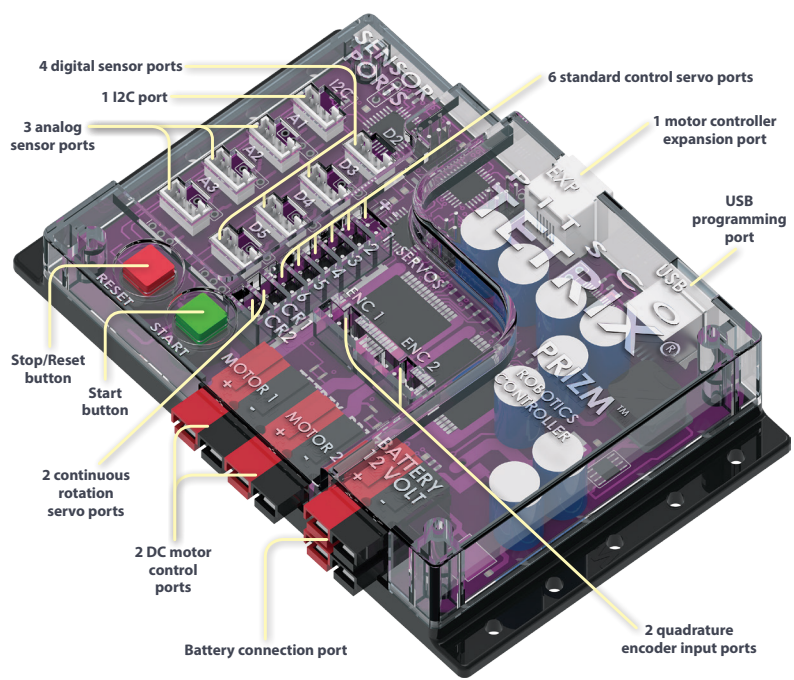
- 1 TETRIX PRIZM Robotics Controller
- 2 Powerpole DC motor cables
- 1 USB cable Type A to B
- 1 battery on/off switch adapter with Powerpole connectors



Technical Specifications:

Microcontroller:	ATmega328P with Arduino Optiboot bootloader installed
Memory:	32 KB flash programmable memory (ATmega328P)
Power:	9-18 volts DC
DC motor ports:	2 Powerpole connections; H-bridge PWM controlled; 10 amps continuous current each channel, 20-amp peak
Recommended motor:	TETRIX® 12-volt DC Motor (39530)
DC motor control modes:	Constant power (-100% to 100%) PID constant speed (-100 to 100 degrees per second) PID constant speed to encoder target position and hold PID constant speed to encoder degrees position and hold Brake or Coast mode when stopping DC motor current monitoring (all modes)
Recommended encoder:	TETRIX MAX Motor Shaft Encoder Pack (38000)
Motor encoder ports:	2 quadrature, 5 volts DC, 50 mA max; Spec: 360 CPR, 1,440 PPR; ENC 1 and ENC 2
USB connector:	USB Type B
USB driver:	FTDI
Standard servo ports:	6 total: servo channels 1-6
Continuous rotation (CR) servo ports:	2 total: CR1 and CR2 channels
Total servo power limit:	6 volts DC, 6 amps max
Servo control modes:	Set servo speed (0% to 100%) Set servo position (0-180 degrees) Set CR servo state (Spin CW/Spin CCW)
Battery voltage monitoring:	0-18 volts range
Digital sensor ports (D2-D5):	Each can be configured as digital input or output. D2 can be configured as a serial communications port.
3 analog sensor ports (A1-A3):	Each can be configured as analog input or digital input/output ports.
1 I2C port (I2C):	100 KHz speed. This connection shares same I2C bus as internal DC motor and servo motor control chips. I2C addresses 0x01-0x06 reserved by the PRIZM Controller.
Motor controller expansion port (EXP):	Additional TETRIX DC motor and servo control modules can be daisy-chained to this port.

Battery connection port:	Powerpole type; additional port for daisy-chaining battery power to motor controllers added to the expansion port
1 green Start button (START)	Programmable push button
1 red Stop/Reset button (RESET)	Non-programmable push button
1 red LED:	Programmable LED used as an indicator
1 green LED:	Programmable LED used as an indicator
1 blue LED:	Indicates the power is on when illuminated
2 yellow LEDs:	Indicates serial data activity on the USB port
1 red and 1 green DC Motor LEDs:	Indicates DC motor rotation and direction for each DC motor channel



Installing the Arduino Software (IDE):

In order to upload programs to the TETRIX® PRIZM™ Robotics Controller, you need to install the *Arduino Software (IDE)*. Full and comprehensive instructions for installing the software on Windows, Mac, and Linux computers can be found at the Arduino website at www.arduino.cc.

To install:

From the homepage, click **Download** to access the software downloads section. After the *Arduino Software* is installed, select **Arduino/Genuino Uno** from the Tools > Board menu to enable communication with PRIZM. You will also need to select the COM port where PRIZM has been installed by your computer system. A more detailed explanation of software installation is included in the full *TETRIX PRIZM Programming Guide* available for download at www.TETRIXrobotics.com/PRIZMdownloads. Here, you will also find the latest updates on supported software platforms for the TETRIX PRIZM Robotics Controller.

Installing the PRIZM Robotics Controller Arduino Library:

After you have successfully installed the *Arduino Software (IDE)*, you can add the PRIZM Robotics Controller Library for Arduino to your programming pallet. The library is a specialized set of functions written specifically for the controller that greatly simplify the creation of programming applications to control your robot's motors and sensors. The PRIZM library is distributed as a zip file: TETRIX_PRIZM.zip.

To install:

Download the PRIZM library from www.TETRIXrobotics.com/PRIZMdownloads. After it is downloaded, save it to a location on your computer so you can install it into the *Arduino Software (IDE)*. If you are unfamiliar with how to install an Arduino library, reference the library installation section in the full *TETRIX PRIZM Programming Guide* that is available for download. You can also visit the Arduino website at www.arduino.cc for instructions on how to install libraries into the *Arduino Software (IDE)*.

Arduino Code Examples:

The PRIZM library includes several code examples that will assist you in understanding how to program PRIZM using the library functions. After the library has been installed, be sure to restart the *Arduino Software*. After the software has restarted, the code examples will be located in the File > Examples drop-down menu.

TETRIX PRIZM Robotics Controller Arduino Library:

Below is a reference of each function statement in the TETRIX PRIZM Robotics Controller Library for Arduino. For a detailed description of each function statement, please download the full *TETRIX PRIZM Programming Guide* found at www.TETRIXrobotics.com/PRIZMdownloads.

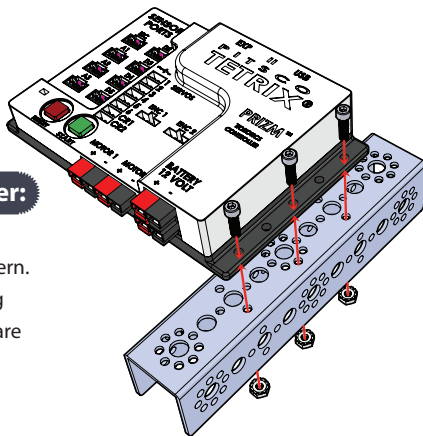
```
PrizmBegin();
PrizmEnd();
setGreenLED(HIGH/LOW);
setRedLED(HIGH/LOW);
setMotorPower(motor#, power);
setMotorPowers(power1, power2);
setMotorSpeed(motor#, speed);
setMotorSpeeds(speed1, speed2);
setMotorTarget(motor#, speed, target);
setMotorTargets(speed1, target1, speed2, target2);
setMotorDegree(motor#, speed, degrees);
setMotorDegrees(speed1, degrees1, speed2, degrees2);
setMotorInvert(motor#, invert);
readMotorBusy(motor#);
readMotorCurrent(motor#);
readEncoderCount(enc#);
readEncoderDegrees(enc#);
resetEncoder(enc#);
resetEncoders();
readLineSensor(port#);
readSonicSensorCM(port#);
readSonicSensorIN(port#);
readBatteryVoltage();
readStartButton();
setServoSpeed(servo#, speed);
setServoSpeeds(speed1, speed2, speed3, speed4, speed5, speed6);
setServoPosition(servo#, position);
setServoPositions(position1, position2, position3, position4, position5, position6);
readServoPosition(servo#);
setCRServoState(CRServo#, state);
```

Sensor Integration:

The TETRIX® PRIZM™ Robotics Controller is designed with sensor ports that are compatible with the Grove system of modular sensors. Currently, the PRIZM Arduino library has integrated support for two of these sensors: the Line Finder and the Ultrasonic. These sensor packs are included in the TETRIX MAX Programmable Robotics Set (43053) and the TETRIX MAX Dual-Control Robotics Set (43054) and are available for purchase à la carte at www.TETRIXrobotics.com. Both à la carte options for the Line Finder Sensor Pack (43056) and Ultrasonic Sensor Pack (43055) include a mounting adapter that aligns with the TETRIX hole pattern, enabling easy robot attachment.

Additional Grove system sensors and Arduino example code for each can be found at www.seeedstudio.com/wiki/GROVE_System. To view the sample Arduino code, simply click the sensor that is of interest.

TETRIX-compatible mounting adapters for the larger family of Grove sensors are sold separately and will be needed for securing sensors to the robot structure. Integrated support for additional Grove sensors will be added to the PRIZM Arduino library as further updates are released.



Attaching the PRIZM Controller:

The PRIZM enclosure mounting holes are spaced to align with the TETRIX hole pattern. PRIZM can be attached to TETRIX building elements using the screw and nut hardware included in the TETRIX robotics sets

Important Safety Information:

Caution: Use only a TETRIX battery pack that is equipped with an in-line safety fuse. Failure to do so could result in damage or injury. Connect the TETRIX battery pack to either the top or bottom red/black power inlet row at the battery connection port. Do **not** connect two battery packs to the PRIZM controller.